Git

# 简介

Git 主张的分布式代码库与文件快照的设计思想，相对于传统 CVS、SVN 等集中式、文件差异式版本控制工具是一种挑战与颠覆。Git 带来了离线提交、轻量级分支等诸多便利。不过，也有人质疑 Git 的复杂性，并由此拔高了学习成本，某种程序上影响了开发者使用或者迁移 Git 的项目进度

**Git 是一个开源的分布式版本控制软件。在英式英语中，Git 指一个愚笨或者不开心的人，恐怕与 Git 发明人——Linux 教父 Linus Torvalds 当时的自嘲心理不无关系吧。2002 年之前，Linux 内核维护工作的绝大部分时间都浪费在提交补丁与保存归档等繁琐事务上。启用版本控制工具 BitKeeper 管理 Linux 内核成了当务之急。不过，BitKeeper 毕竟是一款商业软件，在经历了 3 年免费使用之后，Linux 社区不得不寻求它的替代品，以便继续托管 Linux 内核源代码。2005 年，迫于无奈，Linus Torvalds 自行开发了一套开源版本控制工具，并命名为 Git。**

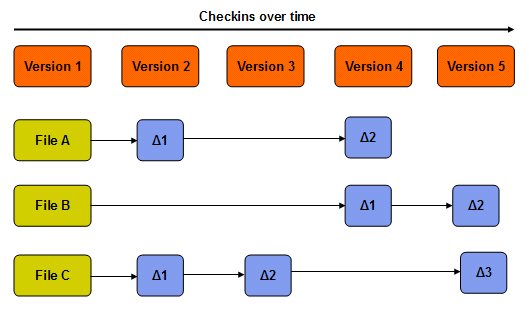
**自诞生以来，Git 就以其开源、简单、快捷、分布式、高效等特点，应付了类似 Linux 内核源代码等各种复杂的项目开发需求。如今，Git 已经非常成熟，被广泛接受与使用，越来越多的项目都迁移到 Git 仓库中进行管理。以 Eclipse 社区为例。据称，目前 80% 的 Eclipse 基金会项目已经完全使用 Git 管理，CVS 访问权限已经切换成只读状态。并且，在 Eclipse 基金会官网中，针对项目管理的介绍中已将"CVS"三个字符划掉，而且很萌地写道，"Ding dong, the witch is dead."，意思是"叮咚，那个老巫婆已经挂了"。**

# 优点

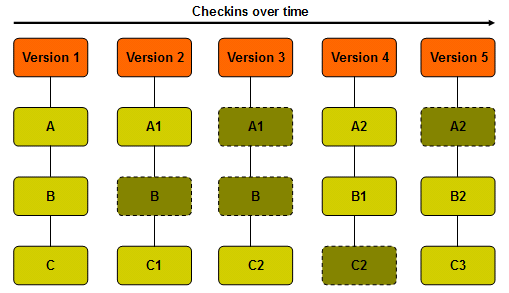
实际上，相对于 CVS、SVN 等主流版本控制软件，Git 的学习成本甚至会更高。比如，对于 Subversion 用户而言，如果能理解什么是文件、工作目录、资源库、版本、分支和标签等概念，差不多就够用了。而对于 Git 用户，需要理解更多更复杂的概念，包括文件、快照、工作树、索引、本地资源库、远程资源库、远程、提交、分支和 Stash 等。那么，为什么软件开发者对 Git 还是趋之若鹜呢？相比于 CVS 与 SVN，Git 的优势到底体现在哪里？

## A:

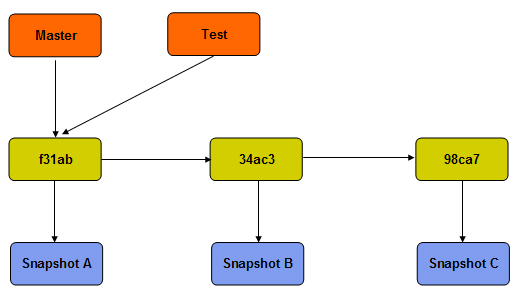
**Git 底层自行维护的存储文件系统是一大亮点。CVS、SVN 底层采用的为增量式文件系统，如图 1 所示。增量式文件系统的特点是：当文件变动发生提交时，该文件系统存储的是文件的差异信息。**

****

同样是文件变更提交，Git 底层文件系统存储的则为文件快照，即整个文件内容，并保存指向快照的索引，如图 2 所示。考虑到性能因素，如果文件内容没有发生任何变化，该文件系统则不会重复保存文件，只是简单地保存文件的链接。

****

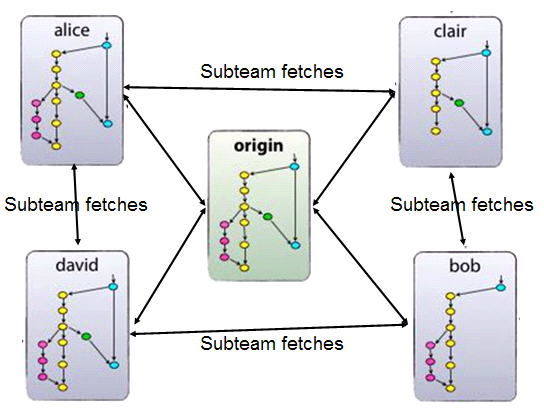
Git 之所以选择这样的底层存储数据结构，主要是为了提高 Git 分支的使用效率。实际上，Git 分支本质上是一个指向索引对象的可变指针，而每一个索引对象又指向文件快照

这样一来，创建分支可以瞬间完成，几乎不需要花费太多代价。换句话说，Git 分支是廉价的、轻量级的。我们看看各种 CVS、SVN 项目，分支通常意味着源代码的完整拷贝，其代价是昂贵的、重量级的。而对于大型项目来说，创建各种分支又是十分必要的，这与 Git 鼓励频繁创建与合并分支的理念相吻合。****

## B:

Git 版本控制系统的设计思想是"去中心化"。传统的 CVS 、SVN 等工具采用的是 C/S 架构，只有一个中心代码仓库，位于服务器端。而一旦由于服务器系统宕机、网络不通等各种原因造成中心仓库不可用，整个 CVS 、SVN 系统的代码检入与检出就瘫痪了。即便考虑到高可用性，通过迁移另一个中心仓库继续代码提交操作，相应的运营维护成本也会随之上升。

为了摆脱对中心仓库的依赖，Git 的初始设计目标之一就是分布式控制管理。我们给出一个样例，如图 4 所示。假如我们成立一个项目组，开发者主要由 Alice、Bob、Clair、David 四名成员组成。其中，除了中心仓库 origin（Git 默认远程仓库名称）之外，每一名成员各自负责一个本地仓库。从分布式的观点来看，David 可看成是 Alice 的远程仓库，反过来也是一样。Git 分布式的设计理念有助于减少对中心仓库的依赖，从而有效降低中心仓库的负载，改善代码提交的灵活性。



Git 分布式设计思想所带来的另外一大好处是支持离线工作。离线工作的好处不言而喻，对于 CVS、SVN 这种严重依赖网络的 C/S 工具而言，没有了网络或者 VPN ，就意味着失去了左膀右臂，代码检入与检出操作就无法正常进行。而一旦使用 Git ，即便在没有 WIFI 的飞机或者火车上，照样可以频繁地提交代码，只不过先提交到本地仓库，等到了网络连通的时候，再上传到远程的镜像仓库。

# 如何安装 Git

总结起来，Git 安装方式通常分为两种：一种是选择 Git 源码编译安装；另一种使用针对特定平台的二进制安装包，又可以细分为 Linux、Mac、Windows 等，其安装说明如下。

## 在 Windows 上安装